第42卷 第5期
2022年10月

桂林电子科技大学学报
Journal of Guilin University of Electronic Technology

Vol. 42, No. 5
Oct. 2022

基于DDQN的多智能体冲突消解方法

张 翼，赵岭忠，翟仲毅

（桂林电子科技大学 计算机与信息安全学院，广西 桂林 541004）

摘 要：针对智能体在局部观测下无法有效决策的问题，提出了一种结合深度强化学习的冲突消解方法。该方法基于DDQN算法，利用强化学习的学习模式的特性，计算智能体的累计回报，通过回报值的大小确定智能体的优先级，从而达到冲突消解的目的。通过模拟现实生活中的堵车场景对该方法进行评估，实验结果表明，该方法能有效解决智能体的冲突。

关键词：多智能体系统；冲突消解；深度神经网络；深度学习；强化学习

中图分类号：TP301　　**文献标志码**：A　　**文章编号**：1673-808X(2022)05-0366-05

A multi-agent conflict resolution method based on DDQN

ZHANG Yi, ZHAO Lingzhong, ZHAI Zhongyi

(School of Computer and Information Security, Guilin University of Electronic Technology, Guilin 541004, China)

Abstract: To solve the problem that agents cannot make effective decisions under local observation, a conflict resolution method combined with deep reinforcement learning is proposed. Based on DDQN algorithm, this method uses the characteristics of reinforcement learning mode to calculate the cumulative return of agent and determine the priority of agent through the return value, so as to achieve the purpose of conflict resolution. The method is evaluated by simulating the traffic jam in real life, and the experimental results show that the method can effectively solve the agent conflict.

Key words: multi-agent system; conflict resolution; deep neural network; deep learning; reinforcement learning

随着人工智能的不断发展，其不断地改变着人们的生活方式和工作方式，当人们在解决现实世界中的问题时，面对越来越复杂、多变的环境，单一智能体已经很难解决，大多需要大量的多智能体协作解决问题。然而，在智能体互相协作的过程中，受到环境、资源等因素的影响，智能体之间难免会产生冲突。例如，如图1所示的交通车辆冲突场景，当多辆智能车要通过同一个路口时，会产生智能体冲突问题，若此时其中一辆车为救护车，救护车必须优先通过路口，则解决这种冲突问题就显得更为迫切。

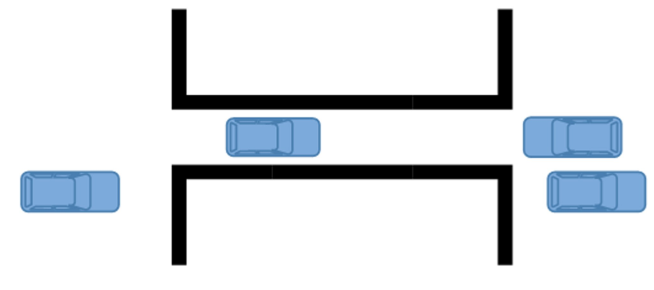


图1　交通车辆冲突场景

近年来，随着强化学习技术的快速发展及其在多个领域的成功应用，各种强化学习方法被应用于多智能体领域[1-3]。因强化学习无需环境建模，智能体能与其所在环境进行自主交互学习，大大提高了计算效率。近期，谷歌团队成功地将强化学习技术与深度学习技术相结合，利用深度神经网络(DNN)来拟合智能体的状态价值函数，解决了智能体维度爆炸的问题[4]。

在单智能体环境中，环境只受单个智能体的影响，因此智能体的局部观察就是对环境的全局观测。但在多智能体环境下，环境受多个智能体的影响，智能体只有自身的局部观测，而无法观测到整个环境，

收稿日期：2022-02-20

基金项目：国家自然科学基金(61862014，61902086)；广西高校中青年教师科研基础能力提升计划(2019KY0249)

通信作者：赵岭忠(1977－)，男，教授，博士，研究方向为软件模型检测，形式化技术。E-mail：zhaolingzhong163@163.com

引文格式：张翼，赵岭忠，翟仲毅. 基于DDQN的多智能体冲突消解方法[J]. 桂林电子科技大学学报，2022，42(5)：366-370.

因而无法有效解决多智能体间冲突问题。部分学者通过在智能体间直接建立通信的方式解决该问题,虽然取得了一定成功,但在有些环境下,却无法建立通信,或通信开销太大,无法适用。

针对以上问题,提出了一种基于 DDQN 的多智能体冲突消解方法,利用标志位对关键信息及状态进行存储,为智能体间建立间接通信,智能体只需通过标志位的信息及自身的局部观测进行学习和决策。智能体根据 DDQN 算法计算出累积回报值,再通过优先级算法得出自身优先级,最后根据优先级进行动作选择。

1　相关知识

1.1　局部可观测马尔科夫决策

在多智能体系统中,各个智能体由于各种因素影响,很难全局观测到整个环境,因此可将整个强化学习过程建模为局部可观测马尔科夫决策过程(decentralized partially observable markov decision process,简称 Dec-POMDP)。通常,将它定义为一个七元组$<N,S,A,P,R,O,\gamma>$,其中:N 为智能体的集合;S 为所有智能体当前时刻的状态及环境信息;$A=\{A_1,A_2,\cdots,A_N\}$表示所有智能体的联合动作集合,A_i 为智能体 i 可选择的所有局部动作的集合;$P:S\times A\times S\rightarrow[0,1]$为状态转移函数,是关于智能体状态和动作的函数,表示在 t 时刻智能体处于状态 s_t,$s_t\in S$ 选择了动作 a,$a\in A$,然后转移到下一状态 s_{t+1} 的概率;R 为所有智能体共享的奖励函数;$O=\{O_1,O_2,\cdots,O_N\}$为所有智能体的联合观测值,O_i 为智能体 i 的观测值;$\gamma\in[0,1]$为折扣因子,表示未来时刻的奖励对累积奖励的影响,是为了避免未来时刻的奖励无限制地叠加而造成累积奖励值无法收敛的情况而设。

在部分可观测马尔科夫决策过程中,智能体无法观测到全局状态,所以智能体只会根据自身的局部观测 O_i 进行决策,从而采取动作;智能体执行动作后,环境会根据状态转移函数转移到下一状态 s',且会收到一个环境反馈的奖励值 r,每个智能体的目标都是最大化累积奖励:

$$G=\sum_{t=0}^{T}\gamma^t r_i^t。$$

1.2　DDQN 算法

DDQN(double deep Q network)[12-14]是对 DQN (deep Q-networks)的一种改进,是为了克服 DQN 中对现实决策的过高估计而提出的。DDQN 结合了深度学习技术,用神经网络拟合智能体的状态价值函数,直接估计智能体的状态和动作值。DDQN 中的内部网络结构与 DQN 相同,但其将智能体动作的选择和对动作价值的评估分别用 2 个神经网络进行训练。神经网络输入的是智能体的观测值,输出的是下一时刻选择的动作值,智能体以一定概率选择最大值的动作作为下一时刻的动作。DDQN 的决策过程如下:

输入智能体的状态信息,网络输出智能体的各个动作价值,也就是 Q 值,DDQN 并不是直接在目标网络中找各个动作估计的最大 Q 值,而是先在当前网络中找出最大 Q 值对应的动作 a,

$$a^{\max}(s'_i,\theta)=\max Q(s'_i,a,\theta),\qquad(1)$$

其中 s'_i 为智能体 i 的下一时刻的状态。通过目标网络获得动作 a 对应的 Q 值,

$$y_j=R_j+\gamma Q'(s'_i,a^{\max}(s'_i,\theta),\theta'),\qquad(2)$$

由式(1)、(2)可得目标 Q 值的计算式:

$$y_j=R_j+\gamma Q'(s'_j,\arg\,\mathrm{ma}Q(s'_j,a,\theta),\theta')。\qquad(3)$$

2　基于 DDQN 的多智能体冲突消解模型

该模型首先利用智能体的状态信息及动作信息计算出智能体在累积时间内的回报值,通过回报值计算出智能体优先级,然后修改标志位。所有智能体根据标志位及各自的局部观测信息进行决策,最终达到冲突消解的目的。

2.1　模型架构及流程

该模型是一个智能体冲突消解模型,其主要思想是利用标志位和智能体的局部观测来计算并得到智能体的优先级,然后进行决策。

智能体内部结构如图 2 所示,由 2 个模块构成,一个是优先级计算模块,该模块主要是通过计算累积步长的环境回报值,并利用回报值来计算智能体的优先级;另一个是动作选取的模块,智能体在得到优先级后,将局部观测值、优先级和标志位信息输入模型,进行动作选择,该模块会对所有可能的动作进行评估,选择奖励值最大的动作作为智能体下一时刻的动作。

整个冲突消解过程主要分为 4 个阶段:

1)智能体通过与环境交互获得局部观测值;

2)利用局部观测值和 DDQN 算法,计算出智能体的累积回报值;

3)将累积回报值输入优先级计算模块,得出智能体优先级,然后修改标志位;

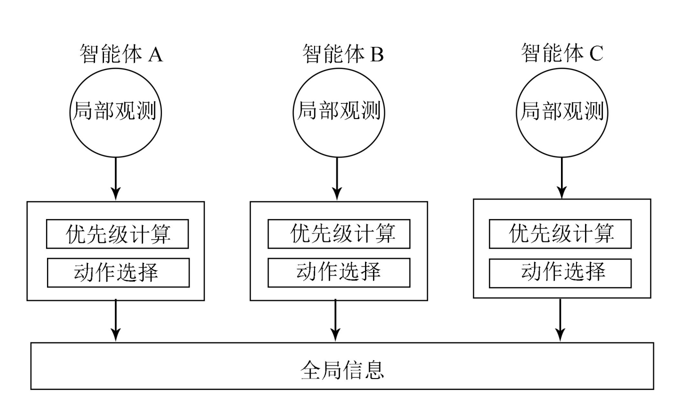


图 2 智能体内部结构

4)将优先级及自身的局部观测值输入动作选择模块中,计算出智能体将要执行的动作。具体模型如图 3 所示。

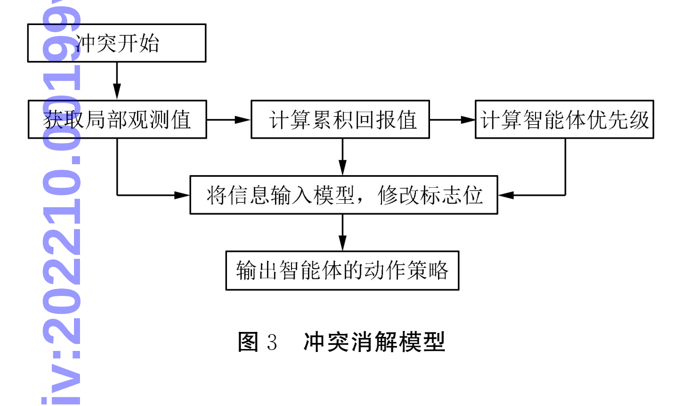


图 3 冲突消解模型

在该模型中,智能体无需对环境进行完整观测,只需根据自身的局部观测和与其余智能体通过标志位进行间接沟通,就能进行有效的决策。

2.2 优先级计算模块

在智能体冲突消解过程中,每个智能体都需要计算出优先级,且智能体之间需要进行协商。因此,针对优先级计算机模块,提出了优先级算法与协商算法。智能体通过优先级算法得出优先级,再利用协商算法得出优先级顺序。

2.2.1 协商算法

协商算法如算法 1 所示,该算法是对智能体间的标志位进行计算修改。具体流程如下:先初始化标志位,然后智能体通过局部观测值计算出优先级;再改变标志位的信息,标志位信息传递给其他智能体;其他智能体在得到标志位信息后,再根据自身的局部观测值计算出新的优先级,最终所有智能体计算得出优先级。

算法 1 协商算法

Input: Agent Number n

Initialize resource flag
for $i=0 \rightarrow n$ do
  Execute A_i
  Set flag
  Send message M_1 to other agent
  Other agent update policy with M_1
  If A_i complete then
    Set flag
  end if
 Send message M_2 to other agent
 update $i+1 \rightarrow i$
End for

2.2.2 优先级算法

优先级算法主要用于计算各个智能体的优先级。

算法 2 优先级算法

Input: learning rate η, mini-batch size k, discount factor γ, network update period r, replay memory D, action-value function Q, weights θ

Output: Network parameter θ
for iteration $=1 \rightarrow M$ do
 for agent $n=1 \rightarrow N$ do
  Sample state s_1
 end for
for step $t=1 \rightarrow T$ do
 for agent $n=1 \rightarrow N$ do
  Select the biggest reward action a_t, n
   with probability ε
  Execute a_t
  Sample state s_{t+1}, and reward r_t
 end for
 Store transition and new message m in D
 for iteration $j=1 \rightarrow k$ do
  Update $\theta \leftarrow \theta + \eta \nabla \theta_j L_j(\theta_i)$
 end for
 update network weight θ with θ_j ever τ step
end for
end for

其中:Q 值是所有智能体的联合 Q 值,在智能体的协作场景中,每个智能体的最佳动作就是各自 Q 值最大的动作;D 是一个公共信息存储位,可被所有智能体计算和存储。该存储空间用于存储各个智能体的状态信息,利用该存储位信息去更新神经网络。智能体以概率 ε 随机选取信息。不同于智能体间直接建立通信的方式[15-20],该方法利用公共标志位为智能体

间建立间接通信,通过存储各个智能体的历史经验信息,智能体只需自身的局部观测和公共的信息就能进行有效决策。

2.3 动作选择模块

在智能体的动作选择模块,神经网络每次都会输出智能体下一时刻可以执行的动作,通常都会选择回报值最大的动作,然后智能体通过选择动作与优先级进行决策,具体过程如下:

通过算法 2 计算出智能体在 t 时刻的累计回报值,

$$R_t^{(i)} = y_{t-1}^{(i)} + \gamma Q'(s'_{t+1}, \theta), \tag{4}$$

其中:s'_{t+1} 为智能体在 $t+1$ 时刻观察到的局部状态;$y_{t-1}^{(i)}$ 为智能体 i 前 $t-1$ 时刻的累计报酬;Q' 为 t 时刻的所选取动作的 Q 值。对于智能体的动作选择,每次选择 Q 值最大的动作,

$$a_t = \max_{a \in A} Q(s_t; \theta), \tag{5}$$

其中,θ 为此时动作选择网络的参数。

根据计算出的个体累积报酬计算智能体的优先级,其中 $V(s_t)$ 为状态值,表示智能体当前状态下的优先级。

$$V(s_t; \theta_e) = E[R_t \mid s_t = s, a_t = (a; \theta_p)]。 \tag{6}$$

3 实验与分析

为了验证和评估基于 DDQN 的智能体冲突消解模型的性能,用公开可用的仿真环境进行仿真实验。方法用 Pytorch 实现,所有的实验均在 64 位 Window 10 电脑上进行,该电脑具有 Intel Xeon E5-1630 CPU@3.10 GHz,8 GiB 内存的配置,且不使用 GPU 加速。此外,用 PyCharm IDE 作为开发环境。

3.1 实验环境

将交通冲突场景构建为一个模拟环境,如图 4 所示。模拟场景的具体规则如下:

1)不同颜色的圆圈代表不同类型的智能体,不同颜色的方格代表不同的目标位置;

2)不同类型的智能体要到达相应颜色的位置才算任务完成;

3)智能体在每个时间步有向上、向下、向左、向右、原地等待 5 个动作可以选择;

4)黑色区域部分是智能体禁止通过的区域,每个时间步智能体只能选择一个动作。

本实验参数设置为:学习率 $\alpha = 0.005$,折扣系数 $\gamma = 0.99$,且折扣系数随着训练的进行逐渐递减,每 300 个步长更新一次神经网络。

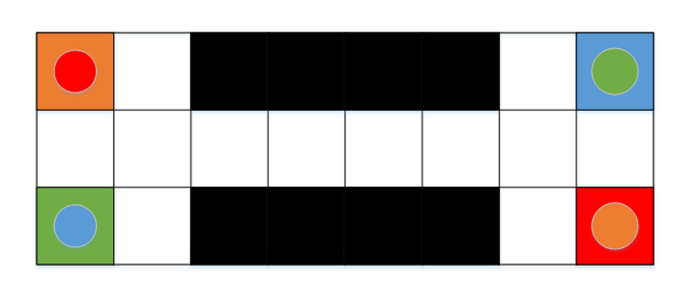


图 4 智能车冲突仿真环境

3.2 实验结果与分析

仿真实验共进行了 50 000 个回合的训练,最终的评价指标为智能体的联合奖励平均值,实验结果如图 5 所示。从图 5 可看出,2 种方法的结果都由开始的快速增长直到趋于稳定,但基于 DDQN 的冲突消解的方法优于传统方法,且该方法能够达到一个较高的回报值水平。

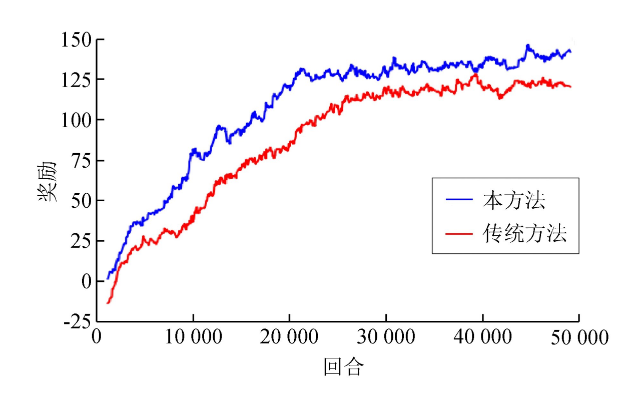


图 5 智能体平均奖励值

对于传统方法,智能体需对整个环境进行建模,存储自身的状态价值信息,增大了计算的复杂性。而对于基于 DDQN 的冲突消解方式,智能体能够自主地与环境进行交互学习,且智能体无需对环境进行全局观察,只需通过标志位及自身的局部观测,就能够进行自主决策。冲突消解的关键在于智能体之间优先级的大小,由于强化学习的特性,可利用智能体的累积回报值计算优先级,因此该方法能较快地得出智能体的优先级。从图 5 还可看出,与传统方法相比,在相同时间内,该方法能使智能体获得更大的回报值,表明该方法能让智能体做出更好的决策。因此,基于 DDQN 的冲突消解模型能更好地解决智能体间的冲突问题,也为局部观测问题提出了新的思路。

4 结束语

对于智能体间的冲突问题,传统方式需要复杂的建模和计算,无法很好地解决。因此,提出了基于

DDQN 算法的智能体冲突消解方法，该方法能够利用智能体的累积回报值快速计算出各个智能体的优先级大小。实验结果表明，该智能体冲突消解方法能有效解决智能体的冲突问题。

参考文献：

[1] TESAURO G. Temporal difference learning and TD-Gammon[J]. Communications of the ACM, 1995, 38(3): 58-68.

[2] KOHL N, STONE P. Policy gradient reinforcement learning for fast quadrupedal locomotion[C]//IEEE International Conference on Robotics and Automation. Piscataway, NJ: IEEE Press, 2004, 3: 2619-2624.

[3] ARULKUMARAN K, DEISENROTH M P, BRUNDAGE M, et al. Deep reinforcement learning: a brief survey[J]. IEEE Signal Processing Magazine, 2017, 34(6): 26-38.

[4] MNIH V, KAVUKCUOGLU K, SILVER D, et al. Human-level control through deep reinforcement learning[J]. Nature, 2015, 518(7540): 529-533.

[5] ZHENG L, YANG Jiacheng, CAI Han, et al. Magent: a many-agent reinforcement learning platform for artificial collective intelligence[C]//Proceedings of the AAAI Conference on Artificial Intelligence. Palo Alto, CA: AAAI Press, 2018: 2-7.

[6] SCHWARTZ H M. Multi-agent machine learning: a reinforcement approach[M]. New York: John Wiley and Sons, 2014: 978-1002.

[7] LOWE R, WU Y, TAMAR A, et al. Multi-agent actor-critic for mixed cooperative-competitive environments[J]. Proceedings of the 31st International Conference on Neural Information Processing Systems. New York, NY: ACM, 2017: 6379-6390.

[8] KAI H, ZHONGHua Z, ZHENG Q, et al. Conflict resolution in multi-agent systems based on negotiation and arbitrage[C]//2010 2nd IEEE International Conference on Information Management and Engineering. Piscataway, NJ: IEEE Press, 2010: 304-307.

[9] GOLPAYEGANI F, DUSPARIC I, TAYLOR A, et al. Multi-agent collaboration for conflict management in residential demand response[J]. Computer Communications, 2016, 96: 63-72.

[10] XIANG Lin, TAO Haijun. Dynamic coalition formation based on multi-sided negotiation[J]. International Journal of Database Theory and Application, 2015, 8(1): 29-38.

[11] MNIH V, KAVUKCUOGLU K, SILVER D, et al. Playing atari with deep reinforcement learning[EB/OL]. (2013-12-19) [2021-08-20]. https://doi. org/10. 4853arxiv. 13125602.

[12] VAN HASSELT H, GUEZ A, SILVER D. Deep reinforcement learning with double q-learning[C]//Proceedings of the AAAI Conference on Artificial Intelligence. Palo Alto, CA: AAAI Press, 2016: 12-19.

[13] RICHARD S, DAVID MC, ALLESTER S, et al. Policy gradient methods for reinforcement learning with function approximation[C]//Proceedings of the 13th International Conference on Neural Information Processing Systems Cambridge, MA: MIT Press., 2000: 1057-1063.

[14] MNIH V, BADIA A P, MIRZA M, et al. Asynchronous methods for deep reinforcement learning[C]//International Conference on Machine Learning. New York, NY: ACM, 2016: 1928-1937.

[15] HAARNOJA T, ZHOU A, ABBEEL P, et al. Soft actor-critic: off-policy maximum entropy deep reinforcement learning with a stochastic actor[C]//International Conference on Machine Learning. New York, NY: ACM, 2018: 1861-1870.

[16] TAMPUU A, MATIISEN T, KODELJA D, et al. Multiagent cooperation and competition with deep reinforcement learning[J]. Plos One, 2017, 12(4): 17-23.

[17] FOERSTER J, ASSAEL I A, DE FREITAS N, et al. Learning to communicate with deep multi-agent reinforcement learning[C]//Advances in Neural Information Processing Systems. New York, NY: ACM, 2016: 2137-2145.

[18] PENG Peng, YING Wen, YANG Yaodong, et al. Multiagent bidirectionally-coordinated nets: emergence of human-level coordination in learning to play starcraft combat games[J]. Arxiv Preprint Arxiv, 2017: 1703-1713.

[19] SAINBAYAR S, ARTHUR S, FERGUS R. Learning multiagent communication with backpropagation[C]//Proceedings of the 29th International Conference on Neural Information Processing Systems, 2016: 5-10.

[20] JIANG J, LU Z. Learning attentional communication for multi-agent cooperation[C]//Advances in Neural Information Processing Systems. New York, NY: ACM, 2018: 7265-7275.

编辑：张所滨